Amendment of Claims

Please amend the claims as indicated in the following listing of claims. This listing of claims will replace all prior versions and listings of claims in the present application.

Listing of Claims

1. (Currently amended) A method for determining ~~software~~ complexity of a software component, comprising the steps of:

~~determining~~creating a plurality of versions of the software component~~whose complexity is to be found~~;

compressing each of the versions, to provide compressed versions;

finding lengths of the compressed versions; ~~and~~

comparing the lengths of the compressed versions; and ~~to provide~~

providing a software complexity metric comprising a comparison of the lengths of the compressed versions.

2. (original) The method of claim 1, wherein the plurality of versions includes raw program text.

3. (original) The method of claim 1, wherein the plurality of versions includes normalized program text.

4. (original) The method of claim 1, wherein the plurality of versions includes normalized unique program text.

5. (original) The method of claim 1, wherein the step of comparing includes a step of finding a ratio using the length of the compressed version of raw program text and the length of the compressed version of normalized program text.

6. (original) The method of claim 1, wherein the step of comparing includes a step of finding a ratio using the length of the compressed version of normalized program text and the length of the compressed version of normalized unique program text.

7. (Currently amended) A method for determining ~~software~~ complexity of a software component, comprising the steps of:

~~determining~~creating raw program text and normalized program text of the software component~~whose complexity is to be found~~;

compressing the raw program text and the normalized program text to provide compressed raw program text and compressed normalized program text, respectively;

finding the length of the compressed raw program text and the length of the compressed normalized program text; ~~and~~

finding a ratio ~~using~~of the length of the compressed raw program text ~~and~~to the length of the compressed normalized program text comparing the lengths of the compressed versions; and ~~to provide~~

providing a software complexity metric comprising the ratio.

8. (Currently amended) A method for determining ~~software~~ complexity of a software component, comprising the steps of:

~~determining~~creating normalized program text and normalized unique program text of the software component~~whose complexity is to be found~~;

compressing the normalized program text and the normalized unique program text to provide compressed normalized program text and compressed normalized unique program text, respectively;

finding the length of the compressed normalized program text and the length of the compressed normalized unique program text; ~~and~~

finding a ratio ~~using~~of the length of the compressed normalized program text ~~and~~to the length of the compressed normalized unique program text; and ~~to provide~~

providing a software complexity metric comprising the ratio.

9. (currently amended) Apparatus for determining ~~software~~ complexity of a software component, comprising:

logic for determining a plurality of versions of the software component~~whose complexity is to be determined~~ and for finding lengths of compressed versions of the plurality of versions of the software;

means for compressing each of the versions ~~of the software whose complexity is to be determined~~, to provide the compressed versions;

means for comparing the lengths of the compressed versions; and ~~to provide~~

means for providing a software complexity metric comprising a comparison of the lengths

of the compressed versions.

10. (currently amended) Apparatus for determining ~~software~~ complexity of a software component, comprising:

logic for ~~determining~~creating raw program text and normalized program text of the software component~~whose complexity is to be determined~~ and for finding ~~the length~~ lengths of compressed raw program text and compressed normalized program text;

means for compressing the raw program text and the normalized program text to provide the compressed raw program text and the compressed normalized program text, respectively; and

means for finding a ratio ~~using~~of the length of the compressed raw program text ~~and~~to the length of the compressed normalized program text; and ~~to provide~~

means for providing a complexity metric comprising the ratio.

11. (currently amended) Apparatus for determining ~~software~~ complexity of a software component, comprising:

logic for ~~determining~~creating raw program text and normalized program text of the software component~~whose complexity is to be determined~~ and for finding ~~the length~~ lengths of compressed raw program text and compressed normalized program text;

means for compressing the raw program text and the normalized program text to provide the compressed raw program text and the compressed normalized program text, respectively; and

means for finding a ratio ~~using~~of the length of the compressed raw program text ~~and~~to the length of the compressed normalized program text; and ~~to provide~~

means for providing a complexity metric comprising the ratio.

12. (currently amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by machine to perform method steps for determining ~~software~~ complexity of a software component, said method steps comprising:

~~determining~~creating a plurality of versions of the software component~~whose complexity is to be found~~;

compressing each of the versions, to provide compressed versions;

finding lengths of the compressed versions; ~~and~~

comparing the lengths of the compressed versions; and ~~to provide~~

providing a software complexity metric comprising a comparison of the lengths of the compressed versions.

13. (original) The program storage device of claim 12, wherein the plurality of versions includes raw program text.

14. (original) The program storage device of claim 12, wherein the plurality of versions includes normalized program text.

15. (original) The program storage device of claim 12, wherein the plurality of versions includes normalized unique program text.

16. (original) The program storage device of claim 12, wherein the step of comparing includes a

step of finding a ratio using the length of the compressed version of raw program text and the length of the compressed version of normalized program text.

17. (original) The program storage device of claim 12, wherein the step of comparing includes a step of finding a ratio using the length of the compressed version of normalized program text and the length of the compressed version of normalized unique program text.

18. (currently amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by machine to perform method steps for determining ~~software~~ complexity of a software component, said method steps comprising:

~~determining~~creating raw program text and normalized program text of the software component~~whose complexity is to be found~~;

compressing the raw program text and the normalized program text to provide compressed raw program text and compressed normalized program text, respectively;

finding the length of the compressed raw program text and the length of the compressed normalized program text; ~~and~~

finding a ratio ~~using~~of the length of the compressed raw program text ~~and~~to the length of the compressed normalized program text; and ~~to provide~~

providing a software complexity metric comprising the ratio.

19. (currently amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by machine to perform method steps for determining ~~software~~ complexity of a software component, said method steps comprising:

~~determining~~creating normalized program text and normalized unique program text of the software component~~whose complexity is to be found~~;

compressing the normalized program text and the normalized unique program text to provide compressed normalized program text and compressed normalized unique program text, respectively;

finding the length of the compressed normalized program text and the length of the compressed normalized unique program text; ~~and~~

finding a ratio ~~using~~of the length of the compressed normalized program text ~~and~~to the length of the compressed normalized unique program text; and ~~to provide~~

providing a software complexity metric comprising the ratio.